Amendments to the Drawings

The attached sheets 1/4, 2/4, 3/4 and 4/4 of drawings includes changes to FIG. 2, FIG. 3 and FIG. 4 and replace the original sheets 1/4, 2/4, 3/4 and 4/4.

In FIG. 2, previously numbered element 10 has been renumbered 11 and conforms to the numbering in the specification. In FIG. 3 and FIG. 4, letters split across two lines have been amended to appear on one line. For example, "$^{AD}_{D}$" has been replaced with "ADD". In FIG 4, line weight and sharding has been conformed to the other figures.

Attachments:

Replacement Sheets 1/4, 2/4, 3/4 and 4/4

Annotated Replacement Sheets 1/4, 2/4, 3/4 and 4/4

Remarks/Arguments

Office Action Summary

Status.

1. This *RESPONSE A* is in answer to the Office communication mailed 07/19/2005.
2. The Office communication is non-final.
3. NA

Disposition of Claims.

4. Claims 1 - 7 are pending in the application.
5. No Claims have been allowed.
6. The rejected Claims 1 - 7 have been amended.
7. NA
8. NA

Application Papers.

9. NA
10. The drawings have been amended.
11. NA

Priority under 35 U.S.C. § 119.

12. NA

DETAILED ACTION

0. Claims 1-7 as amended by this *RESPONSE A* are presented for reconsideration and examination.

0.1 **Drawings.** Alignment errors of text in the drawings have been corrected together with conforming the numbering to the specification.

0.2 **Specification.** The specification has been amended to conform the numbering to the drawings.

Claim Rejections -35 USC § 112

1. The recitation of 35 U.S.C. § 112 is noted.

2. The Examiner rejected Claim 4 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

2.1 The Examiner supports the rejection of Claim 4 as follows:

> *Claim 4 contains the trademark/trade name "S/390". Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See Ex parte Simpson, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark/trade name is used to identify/describe a legacy computer architecture and, accordingly, the identification/description is indefinite.*

2.1.1 The rejection of Claim 4 is traversed for the following reasons. In *Ex parte Simpson,* 218 USPQ 1020 (Bd. App. 1982), the Board was faced with a situation where the applicant did not intend all of the claims at issue "*to be restricted only to products sold under the trademark*". In the present application by way of distinction, applicant's intent is to limit Claim 4 to only the products marketed under the trademark. Further, it is well known that IBM, the trademark owner, has published specifications in great detail defining the goods marketed under the trademark S/390. Through extensive licensing, IBM has publically established a well-known and required technical meaning for the operating systems marketed under the mark S/390. Therefore, "S/390" has obtained both a technical meaning and

a trade mark "source of origin" meaning. The intent of Claim 4 is to adopt exactly the meaning well-accepted in the industry for "S/390". Therefore applicant's use of the trademark S/390 is unlike the use for the goods in *Ex parte Simpson* where the applicant in *Simpson* was intending a broader meaning, not limited exactly to the goods sold under the mark in *Simpson*. Upon reconsideration, it is believed that in the present application, the rule of *Ex parte Simpson* is not controlling and should be distinguished to permit use of the mark S/390.

2.2 The Examiner suggests an interpretation as follows:

The Examiner interprets the term RISC in claim 6 to mean "any architecture other than the legacy architecture", as stated in specifications located on page 3, column 1, paragraph 2, last sentence.

2.2.1 This interpretation to the extent it suggests that "any architecture" is referring to an architecture that has an instruction set that is fully identical to or equivalent to the legacy architecture is traversed. The Examiner is apparently referring to paragraph [0025] of the application as filed and paragraph [0024] of the application as published (US2003/0093649). The meaning of the last sentence of that paragraph must be interpreted in the context of the rest of the paragraph. The term "any architecture" taken in the full context of the paragraph means any architectures having a CISC to RISC relationship.

Claim Rejections -35 USC § 102

3. The recitation of 35 U.S.C. § 102 (b) is noted.

4. The Examiner rejected Claims 1-7 under 35 U.S.C. 102(b) as being anticipated by Danny Ballard (US Patent 4,638,423).

4.1 In making the rejection as to Claim 1 the Examiner argues as follows ("4.1..." numbering added by applicant):

4.1.1 *c. As per claim 1, Ballard discloses a method for dynamic emulation of legacy instructions comprising:*

4.1.2 *i. accessing said legacy instructions in legacy blocks (column: 2 lines: 59-61),*

4.1.3 *ii. for each particular legacy instruction in a particular legacy block, translating the particular legacy instruction into one or more particular translated instructions for emulating the particular legacy instruction (column: 1 lines: 35-38),*

4.1.4 *iii. organizing the particular translated instructions into one or more particular translated blocks (column: 1 lines: 39-43),*

4.1.5 *iv. linking the particular translated blocks into a particular linked group corresponding to said particular legacy instruction (column: 1 lines: 39-43 and 56-62, column: 2 lines: 56-63, column: 3 lines: 22-27}.*

4.1.6 *It is inherent in Ballard's system that to have a working emulator there must be a linking of translated blocks into linked groups corresponding to legacy instructions. Such a task is performed by Ballard's mentioned "control means" for maintaining flow of instructions.*

4.2 The rejection of Claim 1 is traversed for the following reasons.

4.2.1 Ballard has no teaching or suggestion of organizing translated code into translated-code blocks and then linking the translated-code blocks. Ballard allocates memory based upon blocks of target (legacy) instructions and the discussions of blocks in Ballard relates to blocks of legacy instructions, not to blocks of translated instructions as in the present invention.

4.2.2 Each of the references by the Examiner to the Ballard specification in Sections 4.1.2, 4.1.3, 4.1.4 and 4.1.5 above do not mention nor suggest

blocks of translated instructions. All discussion of blocks in Ballard refer to blocks of target (legacy) instructions.

4.2.2.1 Specifically, the Section 4.1.1 and 4.1.2 quotes above are only directed to "legacy blocks".

4.2.2.2 Specifically, the Section 4.1.3 quote above says nothing about blocks in the translated code, to wit, *column: 1 lines: 35-38* of Ballard relied on by the Examiner suggests nothing about blocks in the translated code as follows: *"Yet another object of the present invention is to provide an emulating computer and method of operation that allows direct mapping of a computer instruction set into the instruction set of a target machine."*

4.2.2.3 Specifically, the Section 4.1.4 quote above says nothing about blocks in the translated code, to wit, *(column: 1 lines: 39-43)* of Ballard relied on by the Examiner suggests nothing about blocks in the translated code as follows: *"The above and other objects and advantages of the present invention are provided by an apparatus and method of translating blocks of existing software to software that can be utilized by the emulating computer while the computing process is functioning."*

4.2.2.4 Specifically, the Section 4.1.5 quote above says nothing about blocks in the translated code nor about any block links whatsoever, to wit, *column: 1 lines: 39-43 and 56-62, column: 2 lines: 56-63, column: 3 lines: 22-27* of Ballard relied on by the Examiner suggest nothing about blocks in the translated code as follows:

column: 1 lines: 39-43 , see Section 4.2.2.3 above;

column: 1 lines: 56-62:

The emulating computer comprises: a memory means for storage of instructions and data; a processing means for processing an emulated instruction; a control means for controlling the flow of instructions and data through the emulating computer; and a translating means for translating the instructions.

column: 2 lines: 56-63:

Cache controller 19 monitors address bus 20 and determines if the desired instruction is located in cache RAM 18. If the desired instruction is not in cache RAM 18, cache controller 19 will initiate the transfer of a block of N words of UYK-20 instructions from memory 11 into instruction memory 15. These UYK-20 instructions are then processed, one at a time, through translation unit 17, ...

column: 3 lines: 22-27:

A no match signal line 24 provides the communication of status between cache controller 19 and memory transfer controller 16. If all of the instructions have been translated then memory transfer controller 16 is instructed to move the next block of instructions to instruction memory 15.

4.2.3 Since Ballard has no references to any blocks in the translated code (as is clear from the above analysis in Section 4.2.2 of every reference to Ballard made by the Examiner) the Examiner's argument quoted in Section 4.1.6 that Ballard's system must inherently have "*a linking of translated blocks*" is totally unsupported by anything in Ballard. Ballard has no suggestion that the

translated instructions for a legacy instruction block are distributed over multiple linked blocks of translated instructions. To the contrary, Ballard stores each of the translated instructions in a block large enough to contain the translated instructions as is obvious from FIG 4 of Ballard where the blocks of different sizes per legacy instruction are shown. Such operation in Ballard creates the problem that applicant's invention solves, but Ballard neither recognizes nor solves that problem. Since any particular instruction of Ballard may require a large block, memory allocation for translated instructions in Ballard must be large for each legacy instruction and for each block of legacy instructions. If such large allocation is not made in Ballard, then the Ballard translation cannot be made.

4.2.4 The Examiner's conclusion that Ballard's system must inherently have "*a linking of translated blocks*" is clearly an idea that the Examiner must have wrongfully obtained from applicant's own disclosure since no suggestion thereof whatsoever appears in Ballard.

4.3 In making the rejection as to Claim 2 the Examiner argues as follows ("4.3..." numbering added by applicant):

4.3.1 *d. As per claim 2, Ballard discloses a method of claim 1 wherein*

4.3.1.1 *v. said linking step uses a link in each particular translated block to point to a location of the next particular translated block of the particular linked group (column: 1 lines: 59-61 }.*

4.3.1.2 *This claim is rejected under same reasoning as claim 1 above.*

4.3.1.3 *Additionally, the examiner wishes to point out that it is inherent that Ballard's "control means" have a means for continuing to the next*

instruction, hence would have a link/pointer referring to the next translated block to be executed.

4.3.2 The Examiner's argument quoted in Section 4.3.1.1 is essentially the same as discussed in Section 4.2.2.4 above for Ballard *column: 1 lines: 56-62.* Neither these lines nor anything else in Ballard supports the Examiner's argument.

4.3.3 Claim 2 is traversed for the same reasons as Claim 1 above.

4.3.4 The Examiner's quote of Section 4.3.1.3 finds no support whatsoever in Ballard. There is no basis to conclude that *Ballard's "control means"* has *a means for continuing to the next instruction, hence would have a link/pointer referring to the next translated block to be executed.* Applicant cannot find any suggestion in Ballard for such operation. Quite to the contrary, Ballard appears to store instructions in order in the same block and hence has no need to link from block to block.

4.4 In making the rejection as to Claim 3, the Examiner argues as follows ("4.4..." numbering added by applicant):

4.4.1 *e. As per claim 3, Ballard discloses a method of claim 1 wherein*

4.4.2 *vi. said particular translated instructions are stored in a cache and wherein (column: 2 lines: 56-58}*

4.4.3 *vii. said particular translated instructions are purged from said cache only when all said particular translated instructions of particular translated blocks are also purged from said cache (column: 2 lines: 56-63, and column 2 line 65 to column 3 line 2, column: 3 lines: 22-27).*

4.4.4 *The examiner asserts that according the Ballard the blocks of code would have been moved into memory (according to column 3 lines 22-27) when*

such an action would occur old blocks of code would have been purged from memory.

4.4.5 In making the rejection as to Claim 3, the references to Ballard do not support the Examiner's arguments. The references to *column: 2 lines: 56-63* have previously been discussed above and are repeated here. The references in Ballard to *column 2 line 65 to column 3 line 2, column: 3 lines: 22-27* also do not support the Examiner's argument as follows:

column 2 line 59 (65) to column 3 line 9 (2):

If the desired instruction is not in cache RAM 18, cache controller 19 will initiate the transfer of a block of N words of UYK-20 instructions from memory 11 into instruction memory 15. These UYK-20 instructions are then processed, one at a time, through translation unit 17, which will be discussed in more detail in conjunction with FIG. 2 below, and stored in cache RAM 18. Cache RAM 18 is designed to contain M words where M>N so that when translation unit 19 has processed all of the words from instruction memory 15 there will be a reserve of words in cache RAM 18 while the next block is loaded and translated. Transfer controller 16 will continue to transfer words through translation unit 17 to cache RAM 18 until cache RAM 18 is full. This transfer is accomplished using address bus 21. After the first translated instruction has been stored in cache RAM 18, cache controller 19 will inform MPU 14, along a control line 23, and MPU 14 shall begin execution of the instructions.

column: 3 lines: 22-27:

A no match signal line 24 provides the communication of status between cache controller 19 and memory transfer controller 16. If all of the

instructions have been translated then memory transfer controller 16 is instructed to move the next block of instructions to instruction memory 15.

4.4.6 The citations to Ballard as quoted in Section 4.4.5 do not support the Examiner's argument as quoted in Section 4.4.3 above. Specifically, Ballard does not describe multiple blocks of translated instructions in the Ballard cache. Rather, Ballard insures that the cache has a size *M* that is greater than the size *N* where *N* is the size of a block of legacy instructions. There is no suggestion that the cache in Ballard is organized into multiple blocks where those blocks are linked. Furthermore, the Examiner's conclusion, as quoted in Section 4.4.3, that the citations to Ballard, as quoted in Section 4.4.5 above, refer to purging of the Ballard cache are not accurate. Nowhere does Ballard discuss purging of the cache for any reason or in any manner.

4.4.7 Finally, the Examiner's argument as quoted in Section 4.4.4 that somehow in Ballard "*old blocks of code would have been purged from memory*" is not suggested in anyway by Ballard. Nothing in Ballard discusses purging of the cache for any reason. Clearly, the Examiner must be wrongly relying on applicant's specification to find any discussion of purging.

4.5 In making the rejection as to Claim 4, the Examiner argues as follows ("4.5..." numbering added by applicant):

4.5.1 *f. As per claim 4, Ballard discloses a method of claim 1 wherein*

4.5.2 *viii. said legacy instructions are for a legacy system having a legacy computer architecture (column: 1 lines: 23-26).*

4.5.3 Claim 4 is believed allowable for all of the reasons discussed above in connection with independent Claim 1 from which it depends.

4.6 In making the rejection as to Claim 5, the Examiner argues as follows ("4.6..." numbering added by applicant):

4.6.1 *g. As per claim 5, Ballard discloses a method of claim 1 wherein*

4.6.2 *ix. said legacy instructions are object code instructions compiled/assembled for a legacy architecture (column: 1 lines: 39-43).*

4.6.3 Claim 5 is believed allowable for all of the reasons discussed above in connection with independent Claim 1 from which it depends.

4.7 In making the rejection as to Claim 6, the Examiner argues as follows ("4.7..." numbering added by applicant):

4.7.1 *h. As per claim 6, Ballard discloses a method of claim 1 wherein*

4.7.2 *x. said translated instructions are for execution in a RISC architecture (column: 1 lines: 41-43).*

4.7.3 Claim 6 is believed allowable for all of the reasons discussed above in connection with independent Claim 1 from which it depends.

4.8 In making the rejection as to Claim 7, the Examiner argues as follows ("4.8..." numbering added by applicant):

4.8.1 *i. As per claim 7, Ballard discloses a method for dynamic emulation of legacy instructions, where the legacy instructions are compiled/assembled into object code form for a native architecture, where the legacy instructions are executed as guests in the host architecture, where the legacy instructions are translated to translated instructions in the host architecture and the translated instructions are executed in the host architecture concurrently with the translation of the legacy instructions in the host architecture, comprising:*

4.8.1.1 *xi. accessing said legacy instructions in legacy blocks of a host system operating with said host architecture (column: 1 lines: {4} 46-49, column: 2 lines: 52-56),*

4.8.1.2 *xii. for each particular legacy instruction in a particular legacy block,*

(1) translating the particular legacy instruction into one or more particular translated instructions of the host system for emulating the particular legacy instruction as a guest in said host architecture (column: 1 lines: 35-38),

(2) organizing the particular translated instructions into one or more particular translated blocks (column: 1 lines: 39-43, column: 3 lines: 23-27),

(3) linking the particular translated blocks into a particular linked group corresponding to said particular legacy instruction (column: 1 lines:, 39-43 and 56-62, column: 2 lines: 56-63, column: 3 lines: 22-27).

4.8.1.3 *The examiner asserts that "when the processor attempts to execute an instruction" it is inherently accessing the said instruction.*

4.8.2 All of the references to columns and lines of Ballard in making the rejection of Claim 7 have been discussed above in connection with one or more of the Claims 1 through 6 and are incorpoarated by reference here for Claim 7. Claim 7 is believed allowable for all of the same reasons discussed above.

4.9 In summary, Ballard is a system that translates legacy instructions and stores translated instructions in a cache in sequential order. Ballard has no discussion of the structure within the cache and certainly does not discuss a linked list of blocks for the translated code. Ballard has the problems described in paragraph [0009] of

applicant's specification but Ballard does not recognize those problems or offer any solutions for those problems.

4.10 Further, Claim 1 and Claim 7 have been amended for clarity reciting "*linking the particular translated blocks into a particular linked group corresponding to said particular legacy block.*" Nowhere does Ballard teach or suggest a linked group of translated blocks corresponding to a legacy block.

4.11 For all of the above reasons, Claims 1 through 7 are believed allowable and reconsideration is requested.

5. Information Disclosure Statement (IDS)

5.1 An IDS is submitted herewith.

Respectfully submitted,

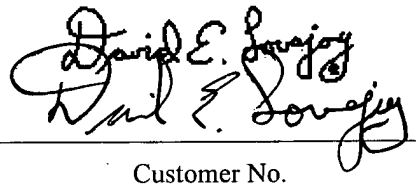
SIGNATURE OF ATTORNEY		
David E. Lovejoy (Reg. No. 22,748)		**Signature Date:** 18 January 2006
Address	Customer No.	Communication
102 Reed Ranch Rd. Tiburon, CA 94920-2025	21603	Tel: (415) 435-8203 Fax: (415) 435-8857 e-mail: david.lovejoy@sbcglobal.net

FIG. 1

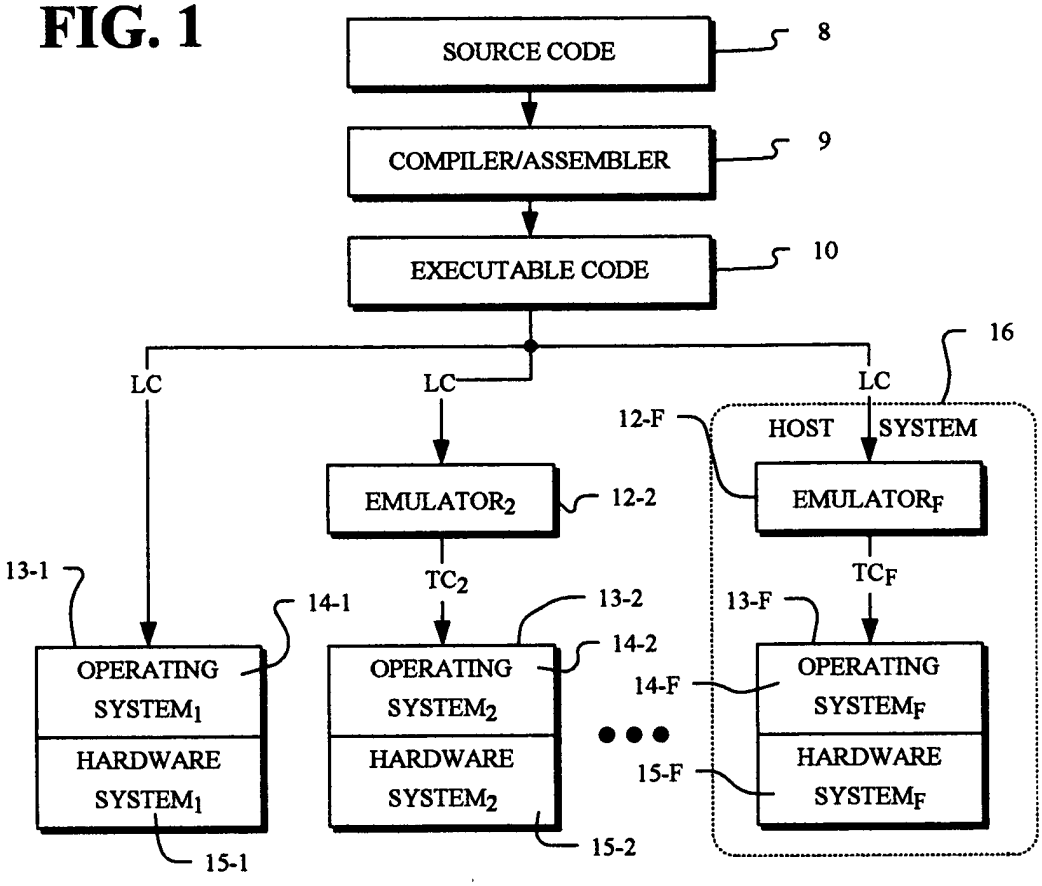
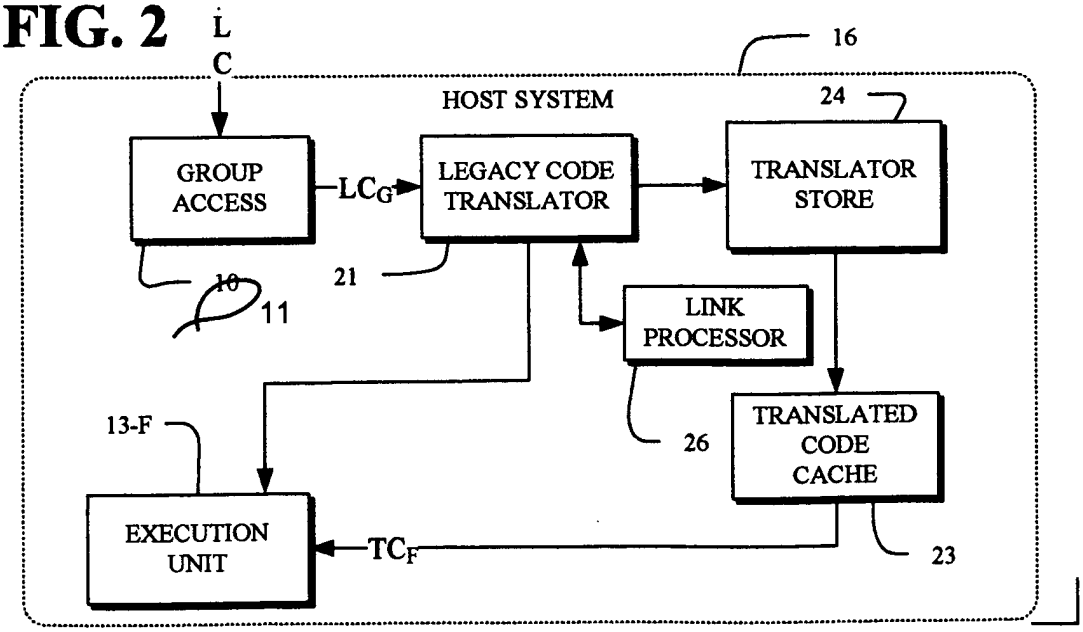


FIG. 2



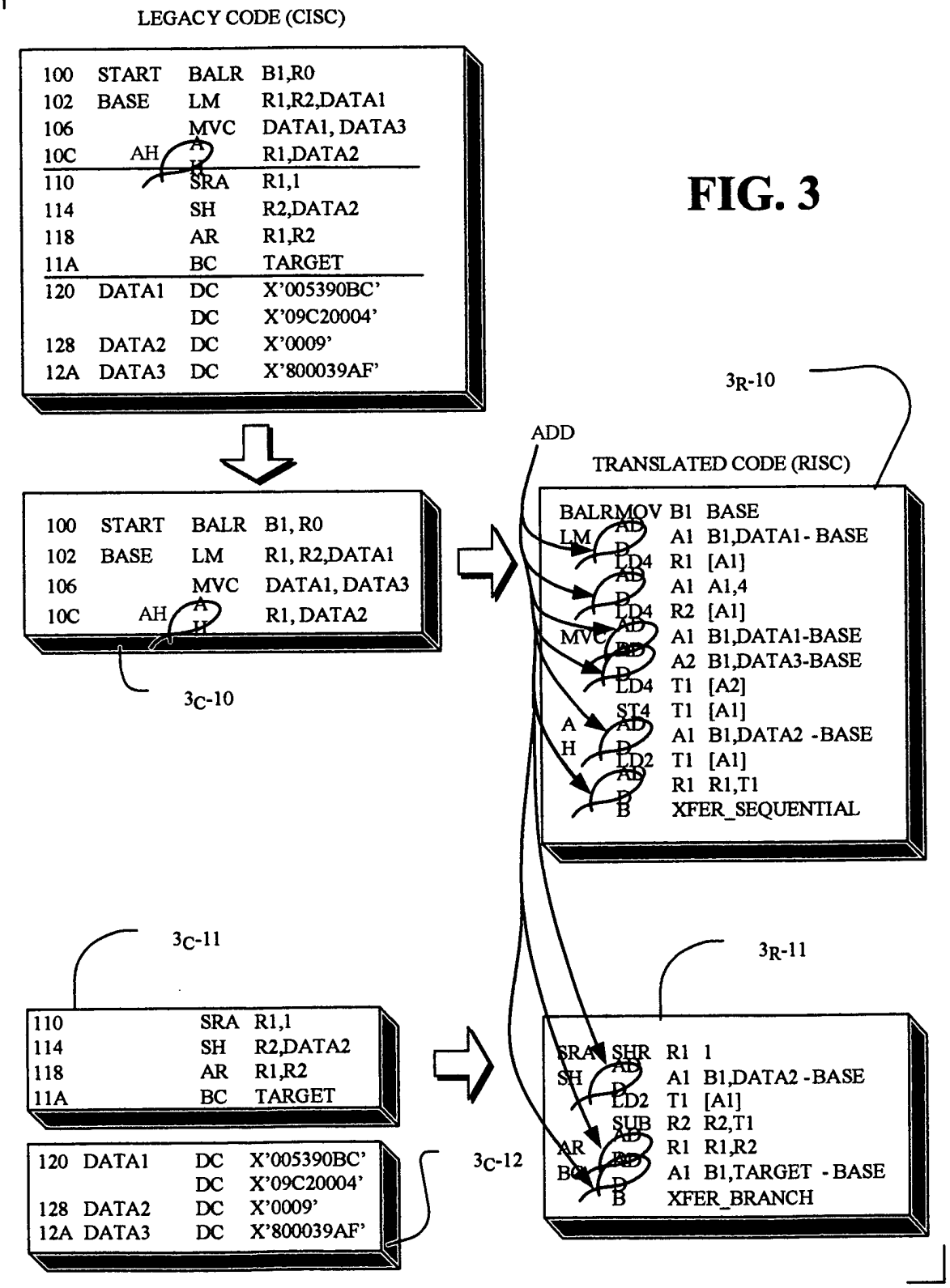

LEGACY CODE (CISC)
100 START BALR B1,R0
102 BASE LM R1,R2,DATA1
106 MVC DATA1, DATA3
10C AH R1,DATA2
110 SRA R1,1
114 SH R2,DATA2
118 AR R1,R2
11A BC TARGET
120 DATA1 DC X'005390BC'
DC X'09C20004'
128 DATA2 DC X'0009'
12A DATA3 DC X'800039AF'
100 START BALR B1, R0
102 BASE LM R1, R2,DATA1
106 MVC DATA1, DATA3
10C AH R1, DATA2
3C-10
3R-10
ADD
TRANSLATED CODE (RISC)
BALR MOV B1 BASE
LM A1 B1,DATA1 - BASE
LD4 R1 [A1]
A1 A1,4
LD4 R2 [A1]
MVC A1 B1,DATA1-BASE
A2 B1,DATA3-BASE
LD4 T1 [A2]
ST4 T1 [A1]
A A1 B1,DATA2 -BASE
H LD2 T1 [A1]
R1 R1,T1
B XFER_SEQUENTIAL
3C-11
110 SRA R1,1
114 SH R2,DATA2
118 AR R1,R2
11A BC TARGET
120 DATA1 DC X'005390BC'
DC X'09C20004'
128 DATA2 DC X'0009'
12A DATA3 DC X'800039AF'
3C-12
3R-11
SRA SHR R1 1
SH A1 B1,DATA2 -BASE
LD2 T1 [A1]
SUB R2 R2,T1
AR R1 R1,R2
BC A1 B1,TARGET - BASE
B XFER_BRANCH

FIG. 3

FIG. 4

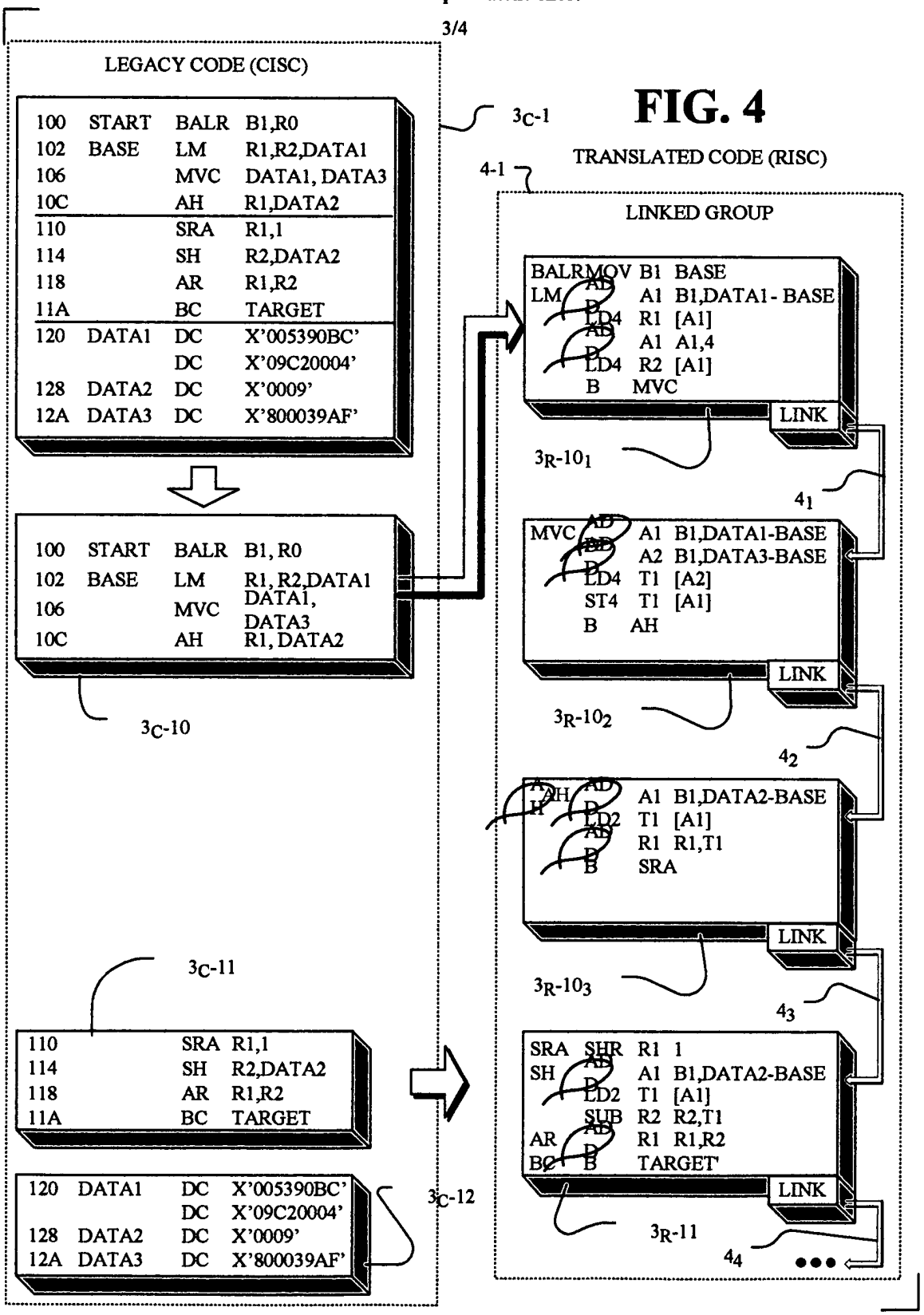
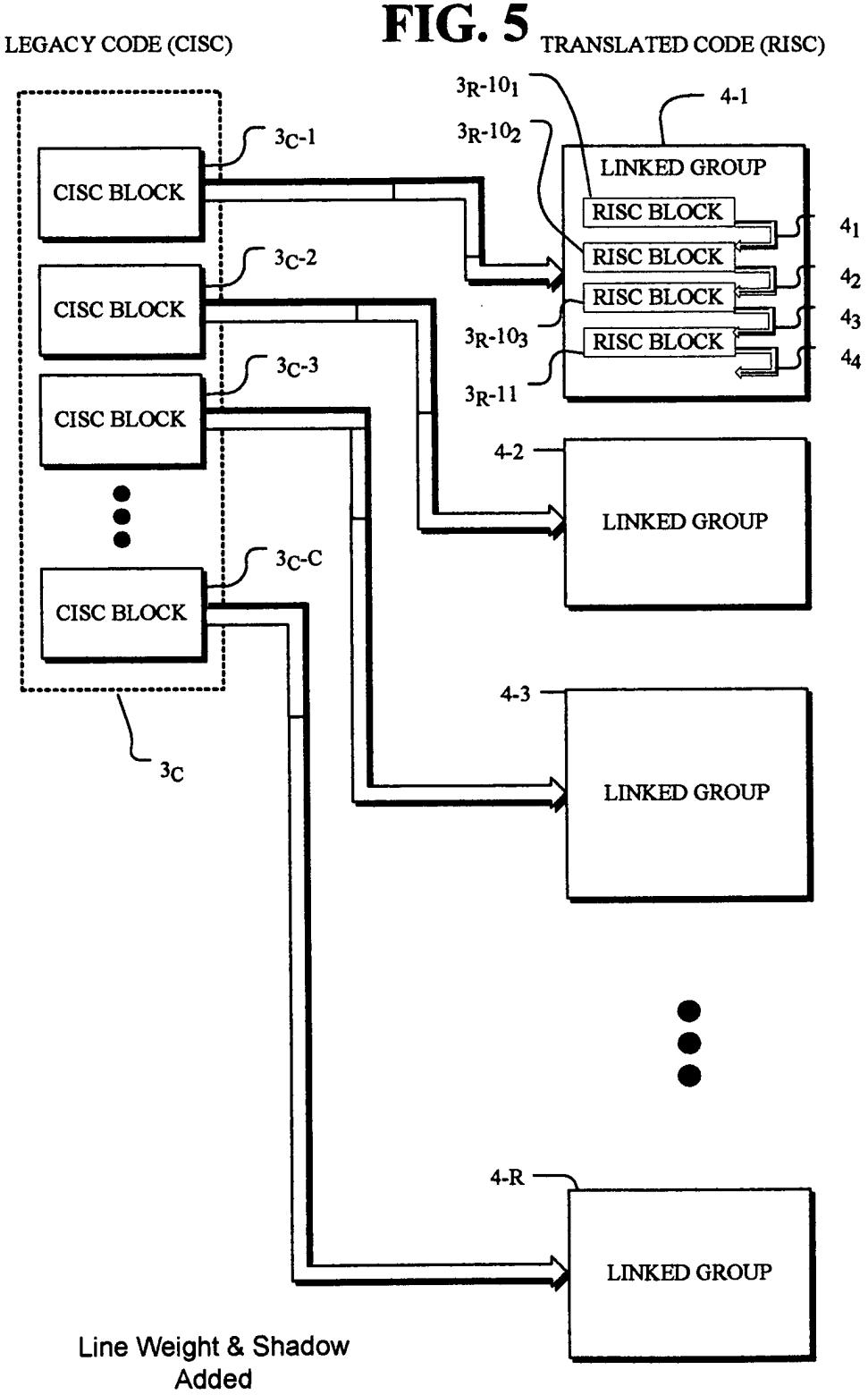

LEGACY CODE (CISC)
100 START BALR B1,R0
102 BASE LM R1,R2,DATA1
106 MVC DATA1, DATA3
10C AH R1,DATA2
110 SRA R1,1
114 SH R2,DATA2
118 AR R1,R2
11A BC TARGET
120 DATA1 DC X'005390BC'
DC X'09C20004'
128 DATA2 DC X'0009'
12A DATA3 DC X'800039AF'
3C-1
100 START BALR B1, R0
102 BASE LM R1, R2,DATA1
106 MVC DATA1, DATA3
10C AH R1, DATA2
3C-10
3C-11
110 SRA R1,1
114 SH R2,DATA2
118 AR R1,R2
11A BC TARGET
120 DATA1 DC X'005390BC'
DC X'09C20004'
128 DATA2 DC X'0009'
12A DATA3 DC X'800039AF'
3C-12
TRANSLATED CODE (RISC)
4-1
LINKED GROUP
BALRMOV B1 BASE
LM A1 B1,DATA1 - BASE
LD4 R1 [A1]
A1 A1,4
LD4 R2 [A1]
B MVC
LINK
3R-101
41
MVC A1 B1,DATA1-BASE
A2 B1,DATA3-BASE
LD4 T1 [A2]
ST4 T1 [A1]
B AH
LINK
3R-102
42
A1 B1,DATA2-BASE
LD2 T1 [A1]
R1 R1,T1
B SRA
LINK
3R-103
43
SRA SHR R1 1
SH A1 B1,DATA2-BASE
LD2 T1 [A1]
SUB R2 R2,T1
AR R1 R1,R2
BC B TARGET'
LINK
3R-11
44

FIG. 5



Line Weight & Shadow Added